# Sequential Churn Prediction and Analysis of Cellular Network Users – A multi-class, multi-label perspective

Farhan Khan and Suleyman S. Kozat
Dept. of Electrical and Electronics Engineering
Bilkent University
Ankara, Turkey
khan@ee.bilkent.edu.tr, kozat@ee.bilkent.edu.tr

***Abstract*—We investigate the problem of churn detection and prediction using sequential cellular network data. We introduce a cleaning and preprocessing of the dataset that makes it suitable for the analysis. We draw a comparison of the churn prediction results from the-state-of-the-art algorithms such as the Gradient Boosting Trees, Random Forests, basic Long Short-Term Memory (LSTM) and Support Vector Machines (SVM). We achieve significant performance boost by incorporating the sequential nature of the data, imputing missing information and analyzing the effects of various features. This in turns makes the classifier rigorous enough to give highly accurate results. We emphasize on the sequential nature of the problem and seek algorithms that can track the variations in the data. We test and compare the performance of proposed algorithms using performance measures on real life cellular network data for churn detection.**

***Keywords*—Sequential learning, Multi-class, Churn, LSTM.**

## I. Introduction

In customer relationship management, retaining the existing users is of utmost importance [1]. Attrition or churn is when a customer leaves an organization or service for the competitor. We seek to analyze cellular customer data for churn analysis and prediction[2]. Generally, a customer would decide to join or leave a network based on several different reasons, for instance, services and their quality, cost, availability of service, customer support and so on. Customer churn is usually a big issue for the organizations specially when key customers decide to leave for a better a competitor service provider [3]. Therefore, we use the current and past customers data to analyze the common grounds for customer attrition. The customer churn prediction helps in identifying and solving the issues that results in attrition. The analysis can be used for possible retention of the current customers.

The cellular customers data contains two different kinds of information or features, i.e., static and sequential. The static features remain constant for a customer for a certain period of time, e.g., gender, age of the customer, start date, etc. The sequential features may vary over time, e.g, usage statistics, bill payments, location,number and type of services etc. Therefore, the network service providers record and keep customer data for every month. We seek to analyze the customers data to identify potential churners.

978-1-5090-6494-6/17/$31.00 

The problem of churn analysis can further be divided into following three aspects.

1) Binary churn detection, i.e., to decide whether a certain customer would churn or not based on his/her available data. This makes it a classical binary classification problem that can be dealt with using machine learning
2) Multi-class churn prediction – to identify in which month a customer would churn in a series of future months.
3) A churn probability analysis to determine the probability of churn of a certain customer in a certain period of time, i.e., a regression problem.

The state-of-the-art classification algorithms, such as the Gradient Boosting Trees, Random Forests etc., are inadequate for churn detection due to the sequential nature of the data since these methods cannot directly incorporate the temporal information [4], [5]. In [6], [7], the authors demonstrate that using a sequential classification algorithm has an improved performance in churn detection when compared to Logistic Regression and Multi-Layer Perceptron. These findings lead to further investigation in sequential learning for boosting the churn detection. Furthermore, the customer dataset needs to cleaned and preprocessed for a more robust analysis [8]. Some further issues with the customers data include missing information, the heterogeneous nature of the data types, i.e., numeric, categorical, binary etc., and the existence of features which are unnecessary or less important as compared to others [8]. Therefore, for a robust analysis we focus on the pre-formatting of the data prior to apply the machine learning algorithms.

The outline of the paper is as follows: In Section II we formally describe the problem setting. Furthermore, we explain the issues and challenges with the real life datasets and their generic solutions. In Section III we introduce our real life cellular network dataset. We clean and preprocess the dataset

and explain the procedure. Once the dataset is ready, we use it for training and validation of several state-of-the-art classification algorithms. We perform several experiments and give the results and comparison using various performance measures, e.g., Receiver Operating Characteristics (ROC) curves and area under these curves (AUC) in Section IV [10]. Finally, we briefly describe our conclusions of our contributions.

## II. Problem Description

All vectors used in this paper are column vectors denoted by boldface lowercase letters. Matrices are denoted by boldface uppercase letters. For a vector $\mathbf{x}$ (or a matrix $\mathbf{U}$), $\mathbf{x}^T$ ($\mathbf{U}^T$) is the ordinary transpose. $M$ is the total number of time-steps and an arbitrary time-step is denoted by $m$ where $0 \leq m \leq M-1$. The time index of a sequence vector is denoted by $m$ in the brackets, as in $\mathbf{x}[m]$. The input vector for a user $n$ is denoted by $\mathbf{x}_n[m]$.

We use two different setting for the input-output relationship in a certain machine learning algorithm, i.e., batch setting and sequential setting. In batch setting, we use the input vector $\mathbf{x}_n$ for cellular network user $n$ where $\mathbf{x}_n \in R^d$, with its element being the features, without considering the temporal nature of the data. In this sense, each user is represented by a $d$ dimensional vector. The desired output is the class label $y$ where $y \in \{0, 1\}$ in case of binary classification, and $y \in \{0, 1, ..., C-1\}$ in a multi-class setting where $C > 2$. We model the desired output as a function of the input, i.e., $y_n = f(\mathbf{x}_n)$.

In the sequential setting, the input $\mathbf{x}_n[m] \in R^d$ is temporal sequence and for user $n$, the input is a matrix $\mathbf{X}_n \in R^{dxM}$ ,i.e.,

$$\mathbf{X}_n = [\mathbf{x}_n[1]\ \mathbf{x}_n[2]...\mathbf{x}_n[M-1]]\,. \tag{1}$$

Here, the rows represent different features and each column corresponds to the time instance. For instance, in a $j * k$ matrix, there are $j$ features and $k$ time instances (months). In sequential setting, the input can be a single column vector, representing the current value of each feature for a certain user, or a $k$ columns matrix containing the data of current and $k-1$ previous months.

We summarize the Churn detection from raw input data as follows:

1) Feature extraction for the users in the training set.
2) Preprocess for categorical feature encoding and sequential features.
3) Batch learning model to train the classifier using known labels.
4) Sequentially update the trained model with new data.
5) Use cross-validation and unlabeled data to verify the model persistence.

In the sequential learning, we use the instantaneous input data, intermediate prediction of the desired label (probability of churn) and the accuracy measure of the decision to train the classifier. We use Keras and Scikit-Learn python libraries in our experiments.

| Features | Description |
|---|---|
| Categorical<br>No. of features = 7 | Gender<br>Device Type<br>Home Change<br>Work Change<br>CRM segment<br>Value segment<br>Lifestyle segment |
| Stationary<br>Numerical<br>No. of feature = 6 | User Age<br>Last reload date<br>Expiry date<br>Hotline date<br>Last reload amount<br>Age of line |
| Sequential<br>Numericals<br>No. of features =25 | Minutes per month with subscription<br>Mintues per month without subscription<br>Duration<br>Payment<br>Outstanding bills<br>internet usage statistics<br>Call drop rate<br>No. of call drops<br>Helpline usage<br>etc. |
| Output Labels | Churn<br>No Churn |

TABLE I: AVEA network dataset description

## III. Datasets

We use the AVEA telecom (now TURK telecom) dataset for our experiments [2]. The dataset consists of sequential features of 6 months. The total number of users are 10000 each having a 40 dimensional feature vector. The first feature, user ID is irrelevant while another feature "tariff type" is constant for all users. The remaining feature set consists of 25 time-varying features, such as the ones related with billing, network usage etc, and 13 stationary features, e.g., device type, age, gender, start date, expiry date etc. Among the stationary features, seven are categorical while the rest are numeric. All the sequential features are numeric. The detailed description of all the features is given in Table I.

### A. Preprocessing and Cleaning

We preprocess the dataset for reshaping to be used for the sequential algorithms, conversion of categorical to numerical variables and imputing missing data. For the batch setting, all the features of a certain users are taken as a $D-$dimensional vector $\mathbf{x}_n$, where $D$ for our dataset is $25 * 6 + 13 = 163$, with 25 sequential features for 6 months and 13 stationary features. In the sequential setting, the features of a certain user $n$ are represented by matrix $\mathbf{X}_n \in R^{M*d}$, where $M = 6$ and $d = 38$. The stationary features are repeated for each month. Furthermore, we convert the categorical features into numeric features using one-hot encoding. Finally, we process the dataset for missing values by inserting the mean of each feature and adding an extra feature to denote the missingness [8]. The final cleaned and processed dataset has 310 features in the batch setting and represented by $72 * 6$ matrix in the sequential setting, i.e., a 72 dimensional vector for each month.

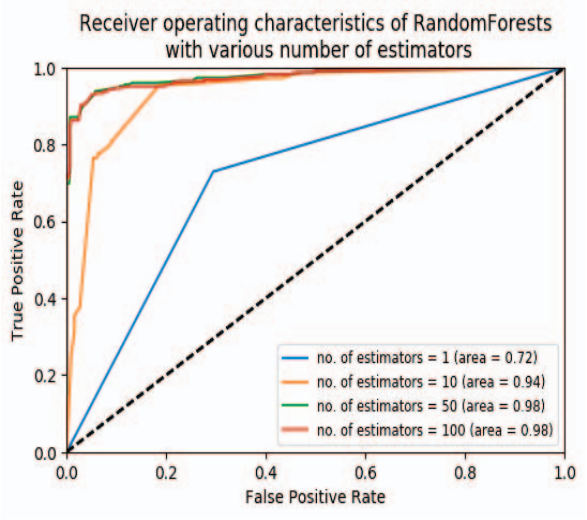


Fig. 1: Receiver Operating Characteristics (ROC) for Random Forest Classifiers

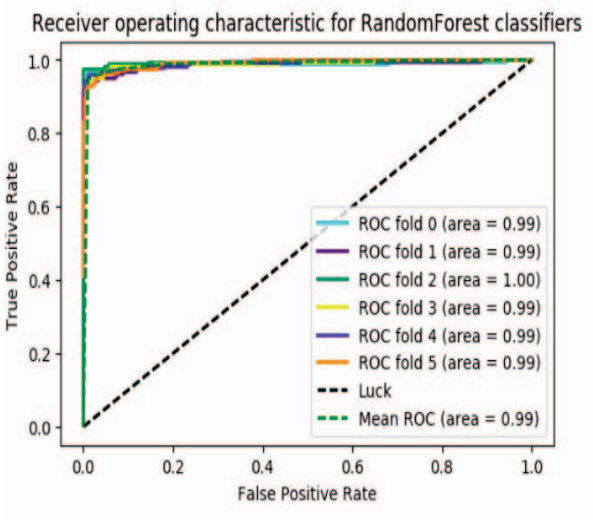


Fig. 2: Receiver Operating Characteristics (ROC) for Random Forest Classifiers with $5-$fold cross-validation

The available output data consists of "Churn" labels for $11$ months including the $6$ month period for which the input data is recorded and further $5$ months in the future. "Churn" is represented by $1$ while "no churn" is represented by $0$. For binary classification, i.e., where the aim is to identify churn status for a certain customer, the output bits are added resulting in $1$ if the user churn during any month and $0$ otherwise. For multi-class churn prediction, the labels are encoded by $12$ distinct classes $\{0, 1, ..., 11\}$, each class number representing the month during which the churn occurred and $0$ representing "no churn".

## IV. Results

In the first set of experiments, we use the AVEA dataset for bi-class churn detection while considering the batch setting. In Fig. 1, we show the receiver operating characteristics of RandomForest classifiers with several choices of number of estimators [11]. We show that as the number of estimators increases, the classifier performance gets better on an unlabeled dataset. However, after using a certain number of estimators, the saturation occurs as in this example, there is no further increase in the AUC scores using more than $50$ estimators. We also use $5-$fold cross-validation for the Random Forest classifier with $50$ estimators and plot the ROC curves for each fold in Fig. 2.

In the following experiment, we compare the performance of several simple and ensemble classifiers namely Support Vector Machine (SVM), Random Forests, Gradient Boost [12], Vanilla RNN and LSTM [9]. For the Vanilla RNN and LSTM, we use the sequential setting as described in Section II and Section III. The input is taken as a $6$ dimensional sequence vector for each feature where each dimension represents a certain month. This makes the complete input dataset as an array of matrices $\mathbf{X}_n$ where $\mathbf{X}_n \in R^{6*72}$. Each user data in the training set is used to train the RNN model sequentially

| Algorithm | Mean AUC scores | Prediction Accuracy | Remarks |
|---|---|---|---|
| SVM | 0.89 | 0.92 | |
| RFC | 0.985 | 0.99 | 50 estimators |
| GBC | 0.98 | 0.99 | 10 estimators |
| VRNN | 0.93 | 0.97 | 3 layers, 32 hidden neurons |
| LSTM | 0.96 | 0.96 | 32 input, 16 hidden layers |

TABLE II: AUC scores and prediction accuracy of Churn predictors

(monthly data). The average AUC scores for each algorithm are shown in Table. II where we use the same batch setting as the previous experiment for the first three classifier models while sequentially training the Vanilla RNN with $60$ neurons at the input and $30$ at the middle hidden layer.

The results in the last two experiments seem promising, however, we assume that all the training data is available in advance (batch). The real and more practical scenario may be different where we are more interested in classification models that are trained instantaneously and sequentially. An adaptive classifier model is able to train on instantaneous input and can incorporate the time varying nature of the dataset and input-output relationship in a more robust manner. Furthermore, sequential algorithms are computationally efficient since they do not necessarily store and process all the previous data.

Finally, we perform experiments to predict the churn probability of a user during each month. Among the $11$ months for which the churn labels are available, the input data is recorded only for the first $6$. Therefore, our goal here is not only to detect the churn in those six months but also predict churn in the next five months based on the history. In this experiment, we consider churn in each month as a separate class that in turns makes it a $12-$class classification problem including the cases when "no churn" is recorded. We use Gradient Boosting

Trees algorithms measure the prediction performance for each class. The results are shown in Fig. 3.

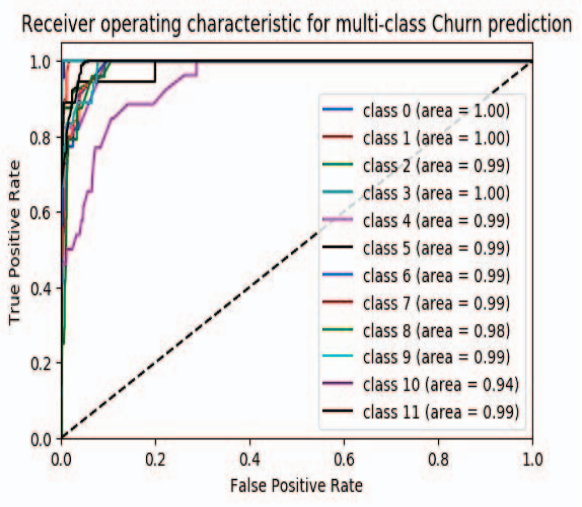


Fig. 3: Receiver Operating Characteristics (ROC) for Random Forest Classifiers

The results in Fig. 3 describe that a optimally trained and tuned ensemble of classifiers can be used to predict the churn of customers well in advance with high accuracy. However, the essential part of learning is correct features selection and suitable manipulation from the users data. Moreover, because of the temporal variation and evolution of the dataset, the previous values can be discarded and in turn make the learning recursive. In addition, several features are linearly dependent on others and a dimensionality reduction is both essential and efficient in terms of performance boost and computations [13]. We compare the prediction performance of Random Forest , GradientBoost and SVM for each of the five future months (the last five months) for which the input data is not available. The AUC scores are shown in Table III.

## V. Conclusion

We consider the issue of customer churn and its prediction from the history and other information available to the service provider from a machine learning and data analysis perspective. We use real life cellular network users dataset and perform a sequential feature set for each user. We describe various type of features ,e.g., categorical, numerical, sequential etc. and analyze their role. The sequential data together with the information of previous churners is used to build and train classification models that can predict the churn probability of customer well in advance. We use several classification algorithm and ensembles and compare the performance of each by using the ROC curves and AUC scores. Our analysis show that we can not only predict whether a customer is prune to churn or not, but also can predict the time of churn well in advance with more than 97% accuracy. The sequential nature of cellular customers data makes this problem a suitable candidate for adaptive and real time learning and prediction.

| Future Month | RandomForest | GradientBoost | SVM |
|---|---|---|---|
| 1 | 0.98 | 0.99 | 0.94 |
| 2 | 0.985 | 0.99 | 0.945 |
| 3 | 1.0 | 0.99 | 0.98 |
| 4 | 0.98 | 1.0 | 0.89 |
| 5 | 0.96 | 0.98 | 0.92 |
| no churn | 0.99 | 0.992 | 0.975 |

TABLE III: AUC scores for monthly churn predictions

## References


[1] W. Reinartz, M. Krafft, W.D. Hoyer, *"The customer relationship management process: Its measurement and impact on performance Journal of Marketing Research, 41 (3)"* pp. 293-305, 2004

[2] F. Khan, I. Delibalta, S. S. Kozat, *"Online Churn Detection on High Dimensional Cellular Data Using Adaptive Hierarchical Trees", EUSIPCO,* 2016.

[3] F. Reichheld, W. Sasser, *"Zero defections: Quality comes to services Harvard Business Review, 68 (5),"* pp. 105-111, 1990

[4] Anita Prinzie, Dirk Van den Poel, *"Incorporating sequential information into traditional classification models by using an element/position-sensitive SAM, Decision Support Systems, 42(2)"* pp. 508–526, 2006.

[5] S. Hochreiter, and J. Schmidhuber, *"Long short-term memory. Neural computation,"* pp 1735 - 1780, 1997

[6] K. Kim, J. Lee *"Sequential Manifold Learning for Efficient Churn Prediction," Expert Systems with Applications, 39(18),* pp 13328-13337, 2012.

[7] J. H. Ahna, S. P. Hana, Y. S. Lee, *"Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry."Telecommunications policy 30(10),*pp 552–568, 2006.

[8] A. Niculescu-Mizil, C. Perlich, G. Swirszcz, V. Sindhwani, and Y. Liu, *"Winning the KDD Cup Orange Challenge with Ensemble Selection,"* Journal of Machine Learning Research Proceedings Track, vol. 7, pp. 23-24, 2009.

[9] K. Greff, R. K. Srivastava, L. Koutnik, R. Steunebrink and J. Schmidhuber, *"LSTM: A Search Space Odyssey," IEEE Transactions on Neural Networks and Learning Systems,* vol.PP, no.99, pp.1–11.

[10] J. A. Hanley, B. J. McNeil, *"The meaning and use of the area under a receiver operating characteristic (ROC) curve," Radiology, 143(1),* pp 29–36, 1982.

[11] Breiman, *Random Forests, Machine Learning, 45(1),* pp 5–32, 2001.

[12] J. Friedman, *Greedy Function Approximation: A Gradient Boosting Machine, The Annals of Statistics,* Vol. 29, No. 5, 2001.

[13] F. Khan, D. Kari, I. A. Karatepe, S. S. Kozat, *"Universal Nonlinear Regression on High Dimensional Data using Adaptive Hierarchical Trees", IEEE Transaction on Big Data,* Vol. 2, No. 2, pp. 175–188, June 2016.